WarpPCX

Oliver Roberts

COLLABORATORS

	TITLE : WarpPCX		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Oliver Roberts	January 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 WarpPCX 1

1.1 WarpPCX.datatype 45.3 . . . 1

1.2 Description . . . 2

1.3 Disclaimer . . . 2

1.4 Distribution Conditions . . . 2

1.5 Registration . . . 3

1.6 Features . . . 4

1.7 System Requirements . . . 4

1.8 Installation . . . 5

1.9 Preferences Options . . . 5

1.10 Speed . . . 6

1.11 Acknowledgements . . . 8

1.12 Support & the Future . . . 8

1.13 About the author . . . 9

1.14 Program History . . . 9

Chapter 1

WarpPCX

1.1 WarpPCX.datatype 45.3

```
                WarpPCX.datatype 45.3 - a
            fast
             24-bit capable PCX picture datatype

            SHAREWARE
                      (for 68k, PPC/OS4, PPC/WarpOS and PPC/MorphOS)
```

Description
what is this datatype for?

Disclaimer
important notices

Distribution
distribution conditions

Registration
registering this Shareware software

Features
list of features

System requirements
what you need to use this software

Installation
installing this software

Preferences
descriptions of configurable settings

Speed
information regarding speed issues

Acknowledgements
thankyous and credits

Support & the Future
support + improvements I intend to make

About the author
how to contact the author

History
program history

1.2 Description

WarpPCX.datatype is a datatype which allows you to view and read ZSoft PC Paintbrush (PCX) image files. A unique feature of this particular PCX datatype is that in addition to supporting the 68k (with optimized versions for 020, 030, 040 and 060), it also supports the PowerPC processor, with native OS4, WarpOS and MorphOS versions. Even better, it is fast, compact, clean and well behaved - a true plug'n'play PPC datatype. One of the key features is its superior speed, hence the name WarpPCX. This datatype uses the same engine as used by my other WarpDTs.

1.3 Disclaimer

This software is provided "as is", without warranty of any kind, either expressed or implied, statutory or otherwise. By using the archive and its contents, you accept the entire risk as to its quality and performance.

Neither Oliver Roberts nor any other party involved in the creation, production or delivery of the archive and its contents shall be liable for any direct, indirect, special, consequential or incidental damages, including without limitation damages for loss of profits, loss of use or loss of anticipated costs, expenses or damages, and any data or information which may be lost or rendered inaccurate, even if Oliver Roberts is advised of the possibility of such damages.

Do not attempt to tamper with the supplied files. Doing so will cause problems and you may find things start going wrong!

1.4 Distribution Conditions

WarpPCX.datatype is shareware, with the copyright remaining with the
author
and may be freely distributed legally providing:

(1) None of the distributed files are changed in any way
(2) It is not sold for profit and it is not included on any disks that are sold solely for profit (includes magazine coverdisks)
(3) The distribution contents remain complete (see list below)
(4) WarpPCX.datatype is registered if it is found to be useful after the evaluation period
(5) Registered user keyfiles are NOT distributed along with copies of WarpPCX.datatype

If this software is to be sold for profit, permission must be obtained from me, the author .

Aminet and Amigactive have been granted permission to distribute WarpPCX.datatype on their CDs.

The following files must be present in their original and unchanged form in any copies of this software:

```
   Classes/Datatypes/WarpPCX.datatype.020
   Classes/Datatypes/WarpPCX.datatype.030.pch
   Classes/Datatypes/WarpPCX.datatype.040.pch
   Classes/Datatypes/WarpPCX.datatype.060.pch
   Classes/Datatypes/WarpPCX.datatype.mos
   Classes/Datatypes/WarpPCX.datatype.os4
   Classes/Datatypes/WarpPCX.datatype.wos
   Devs/Datatypes/PCX
   Devs/Datatypes/PCX.info
   WarpPCX.guide
   WarpPCX.guide.info
   Install_WarpPCX
   Install_WarpPCX.info
   spatch
```

1.5 Registration

WarpPCX.datatype is part of the WarpDT package, and is SHAREWARE, which means you are expected to register if you find it useful, and wish to continue using it. I have decided not to cripple the datatype at all during the evaluation period, so everybody can judge for themselves if this product is worth buying.

Registration of the WarpDT package costs £15 (GBP), entitling you to unlimited usage of all the WarpDTs for all CPU/OS versions. At this time, this one-time registration fee entitles you to full versions of any new datatypes added to the WarpDT family in the future.

To register, please visit http://www.warpdt.co.uk/ for further instructions.

1.6 Features

```
 - Read standard ZSoft PC Paintbrush PCX files, including less common
   non-standard modes which Cloanto Personal Paint can save

 - Supports pixel bit depth of 1 or 8 and 1-8 planes of data, covering
   mono 1-bit images upto 24-bit RGB, both RLE compressed and uncompressed

 - Highly optimized datatype dispatch engine and PCX decoder,
   resulting in a very efficient, compact and quick PCX datatype

 - Asynchronous double buffering techniques (WarpOS only) and file i/o,
   which speeds up image decoding

 - Optimized versions for 68020, 030, 040 and 060

 - PowerPC support with native OS4, WarpOS and MorphOS versions

 - Alter the pen allocation precision when images are remapped to an
   8-bit display

 - Specific support for the AmigaOS picture.datatype v44+, when available

 - The dithering feature of the AmigaOS picture.datatype v44+ can be
   configured to your liking (e.g. disabled for 15/16-bit displays)
```

1.7 System Requirements

```
This datatype needs the following in order to work:

 - Kickstart 3.0 or higher
 - picture.datatype v43 or higher
   (i.e. either of the ones supplied with AmigaOS 3.5+, P96 or CGraphX)

 68k version

 - 68020 processor or higher (optimized versions included)

 WarpOS version

 - PPC accelerator card + 68040/060
 - WarpUp Release 5.0 or higher

 OS4 version

 - PPC accelerator card / AmigaOne
 - AmigaOS 4.0 pre-release or newer

 MorphOS version

 - PPC accelerator card / Pegasos
 - MorphOS beta release 1 or higher

Note that a graphics card is not necessary.
```

1.8 Installation

To install WarpPCX.datatype, simply run the provided installer script by double-clicking the icon. This will create a CPU specific version of the library for you, and install it to SYS:Classes/Datatypes and also installs the PCX descriptor file to DEVS:Datatypes. It does nothing else, so you needn't worry about it messing about with your system :)

1.9 Preferences Options

WarpPCX can be configured using the "Datatypes/WarpPCX.prefs" environment variable. The preferred way of altering the settings is via the WarpDTPrefs graphical user interface (requires OS3.5 or higher), available from http://www.nanunanu.org/~oliver/warpdtprefs.html or Aminet (util/dtype/WarpDTPrefs.lha).

However, it is also possible to alter the settings manually by changing the environment variable with setenv, according to the following template:

```
  DITHER_OVERRIDE/K,DITHER_QUALITY/K,DITHER_DEPTH/N/K,PENS_OVERRIDE/K,
  PENS_QUALITY/K
```

OS3.5 DITHER CONTROL OPTIONS

The three dither options all interact with each other and apply only when at least picture.datatype V44 (OS3.5/3.9) is in use.

By default picture.datatype V44+ dithers images for all target displays less than 24-bit. It is debatable whether it's worth using dithering on 15/16-bit displays - for photographic images, the difference is usually not noticable, but for computer generated images containing smooth colour gradients dithering produces a noticably better output image.

DITHER_OVERRIDE

Selects what our dither quality setting should override:

```
  NOTHING    - do not change the dither quality in any circumstances
  DEFAULTS   - only override picture.datatype's default dither quality,
               which is used when the application does not specify it's
               own dither quality value
  APPS       - override the dither quality that an application may set
  EVERYTHING - always use your custom dithering settings
```

DITHER_QUALITY

Select the dither quality to use when this datatype decides that changes to the dither quality are required, according to your dither override and depth settings:

```
  POOR - simple colour remapping and no dithering (fastest)
  GOOD - slower, higher quality dithered output (default)
  BEST - similar to GOOD, with maybe slightly higher quality output
```

DITHER_DEPTH

Use this option to select which display depths should use the above quality setting. When the target display is deeper than the given depth value, the above dither quality will be applied to that image. For example setting this to 8 will cause the dithering settings to be used for 15/16-bit target displays, but left untouched when the target display is <= 8-bit.

PEN SELECTION OPTIONS

The pen selection options are only relevant to 8-bit displays, and adjust the precision to which pens/colours are allocated, which affects the image quality and number of pens that will get used.

PENS_OVERRIDE

Selects what our pen/colour quality setting should override:

```
  NOTHING    - do not change the pen precision under any circumstances
  DEFAULTS   - only override picture.datatype’s default pen precision
               quality, which is used when the application does not
specify
               it’s own pen precision value
  APPS       - only override the pen precision that an application has set
  EVERYTHING - always use your pen precision settings
```

PENS_QUALITY

Changes the pen precision quality to use when remapping an image to an 8-bit display, in situations according to your pens otherride setting:

```
  POOR - use only a small amount of unique colours, at the expense of
         image quality
  GOOD - allocate enough pens to ensure a reasonable output image
         quality, whilst not hogging all pens for a single image (default)
  BEST - allocate as many pens as possible/necessary, resulting in the
         best image quality, but when displaying multiple images on the
         same screen, the quality of every image might not be so good
```

1.10 Speed

A fast PCX datatype?

The table below shows the time (in seconds) it took to decode 5 different images on my A1200 603e/240MHz 060/50MHz, with BVision and CGX picture.datatype, with other tested datatypes configured as close as possible to WarpPCX’s internal settings. However, due to the nature of PCX files, don’t expect miracles from the PPC versions.

Really, there wasn’t much to benchmark my datatype against, apart from the CGX PCX datatype. And, as you can see, there’s really not a lot in it in terms of speed. The main reason being that PCX files use a very simple RLE compression scheme, which in turn means simpler decoding routines, and therefore less room for optimisation. But, WarpPCX tends to be faster than everything else, even if only small fractions of a second :-)

	PowerPC		
	WarpOS	MorphOS	
	WarpPCX 45.3		CGX PCXdt 44.2
833x841 1-bit c, 87019 bytes	0.19	0.16	0.22
621x494 4-bit c, 153035 bytes	0.21	0.21	0.38
461x599 24-bit c, 62967 bytes	0.15	0.15	0.15
645x259 8-bit c, 130842 bytes	0.11	0.11	0.11
268x577 8-bit c, 23715 bytes	0.08	0.06	0.08

	68K		
	WarpPCX 45.3	CGX PCXdt 43.7	C= PicDT 42.1
833x841 1-bit c, 87019 bytes	0.13	0.15	0.89
621x494 4-bit c, 153035 bytes	0.26	0.30	n/s
461x599 24-bit c, 62967 bytes	0.28	0.33	n/s
645x259 8-bit c, 130842 bytes	0.14	0.16	n/s
268x577 8-bit c, 23715 bytes	0.07	0.09	0.37

(c = compressed, n/s = not supported)

Note: all tests were performed, multiple times, using Visage, with the following command line: "visage test.pcx time test".

WarpOS version too slow!

The main problem is that PPC datatypes still have to use the 68k for reading the data from disk and for creating / writing to the bitmap structures that the datatypes system requires. As far as the former goes, time lost for file i/o is negligable as WarpPCX uses double buffered asynchronous i/o (supports DMA controllers).

The largest bottleneck is that the DTM_WRITEPIXELARRAY method of the picture.datatype has to be used to write the image data from WarpPCX into the image bitmap. As this process is done via picture.datatype, it can only currently be performed by the 68k. To give you some idea of how much of a problem this is for WarpPCX, typically, half of the overall decode time is used by the PPC to decode the whole image, and the other half is used by DTM_WRITEPIXELARRAY on the 68k. And that's on a graphics card - the time used by DTM_WRITEPIXELARRAY will probably be even greater on systems using native Amiga graphics. It doesn't take a genius to see that this is slowing the datatype down, and is the main reason why WarpPCX will still be faster on a 060 than a 040.

How to make the datatype faster

Is there anything that can be done about this? Well, yes, there are a few patches that you can install which should make things faster:

- NewWPA8 (util/boot/NewWPA8.lha on Aminet) should provide a notable speed increase on native Amiga graphics - probably won't make any difference if you use a graphics card.

- If you use a graphics card and CyberGraphX, you may want to make sure you are using the supplied v43 picture.datatype, as this will be faster than the P96 and OS3.5/3.9 picture.datatype on your system.

Of course, any other general speed-up patches should help too.

1.11 Acknowledgements

The PCX decoder routines were written from scratch in order to suit my WarpDT engine, providing maximum performance. The ZSoft PCX technical reference manual, John Bradley's XV and GIMP were used as a reference. Cloanto's Personal Paint was also used to create some more unusual PCX files.

The WarpOS version was made possible by VBCC, which was used to build and compile the datatype. Thanks to Volker Barthelmann and Frank Wille, for their support and help.

The Spanish installer translation is by Dámaso D. Estévez. The French installer translation is by Philippe Bovier.

Thanks also to Sam Jordan for WarpOS and helping me out with various queries regarding it.

Finally, thanks to the OS 3.5 development team - now everyone has access to a 24-bit picture.datatype, I don't need to bother messing about adding dithering routines :)

1.12 Support & the Future

Some things that may appear in the future:

- If I can squeeze any more speed out of the datatype, in general, I'll do so :)

If you have any other suggestions, please
let me know
.

Future releases of WarpPCX.datatype will be available from either Aminet (util/dtype/WarpPCXdt.lha) or its webpage:

http://www.warpdt.co.uk/pcx.html

```
If you would you like to know when WarpPCX is next updated, then you may
want to subscribe to my announcement list to receive an e-mail informing
you of the changes as soon as a new versions of any of my products are
released.  To subscribe, send a blank e-mail to

  futaura-announce-subscribe@yahoogroups.com

or go to

  http://groups.yahoo.com/subscribe/futaura-announce
```

1.13 About the author

```
If you have any problems with this software, or if you have any
suggestions/queries, please contact me and I will do my best to sort any
bugs out as soon as possible:

e-mail: oliver@futaura.co.uk
   www: http://www.nanunanu.org/~oliver/
   icq: 34640231
```

1.14 Program History

```
45.3 (4.8.2004)

 - Fixed bug that prevented the 68k and WarpOS classes from being flushed
   from memory.

45.2 (30.6.2004)

 - Added native AmigaOS 4 version.
 - Fixed memory leak that affected all versions.
 - Fixed bug in WarpOS version which could cause the datatype to crash,
   especially with very small images.
 - Added Polish installer translation.
 - Minor changes and improvements.

45.1 (25.8.2003)

 - Initial release.
```